(12) UK Patent Application (19) GB (11) 2 114 407 A

(21) Application No 8302122
(22) Date of filing 26 Jan 1983
(30) Priority data
(31) 8202551
(32) 29 Jan 1982
(33) United Kingdom (GB)
(43) Application published
17 Aug 1983
(51) INT CL^3
G06F 3/153
(52) Domestic classification
H4T 4R BRB
(56) Documents cited
None
(58) Field of search
H4T

(71) Applicants
National Research
Development
Corporation,
(Great Britain),
101 Newington
Causeway,
London SE1 6BU.

(72) Inventors
Andrew David
Lambourne,
Robert George Baker,
Alan Francis Newell.

(74) Agent and/or Address for
Service
V. Hasler,
Patent Department,
National Research
Development
Corporation,
101 Newington
Causeway,
London SE1 6BU.

(54) Methods and apparatus for use in arranging text

(57) Subtitling for example for television is time consuming because it is necessary to divide many subtitles into lines and for maximum intelligibility the ends of lines must be carefully selected. In addition for Teletext it is required to box each subtitle and to colour the text and background. Apparatus and method are described for entering subtitle next using a keyboard 10 into a computer 11 where the syntax and punctuation of the text is analysed and weights dependent thereon are inserted into a stored version of the text. The computer determines suitable ends for subtitle lines on the basis of the weighting and the effect of the position of possible line ends on the shape of the subtitle. Finally the subtitle is stored in an output buffer of the computer together with characters controlling the beginning and end of each line, its colours and position and then the contents of the buffer are recorded or transmitted if 'live'.

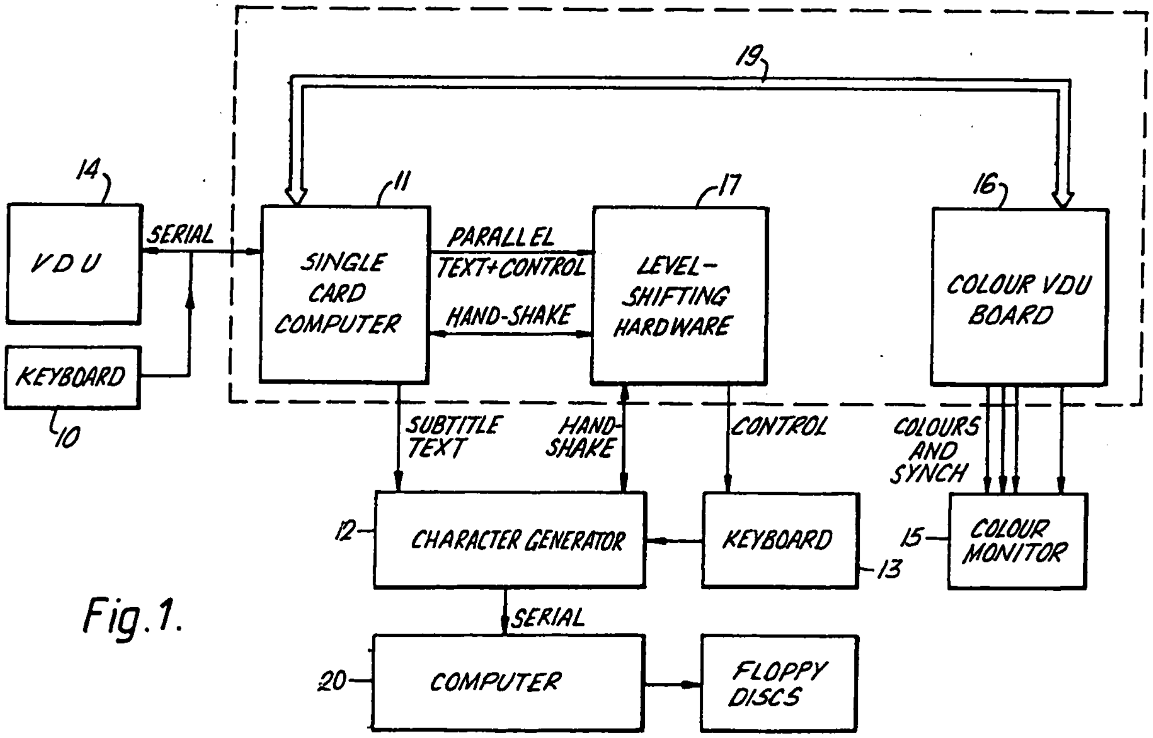


Fig.1.

GB 2 114 407 A

1/12

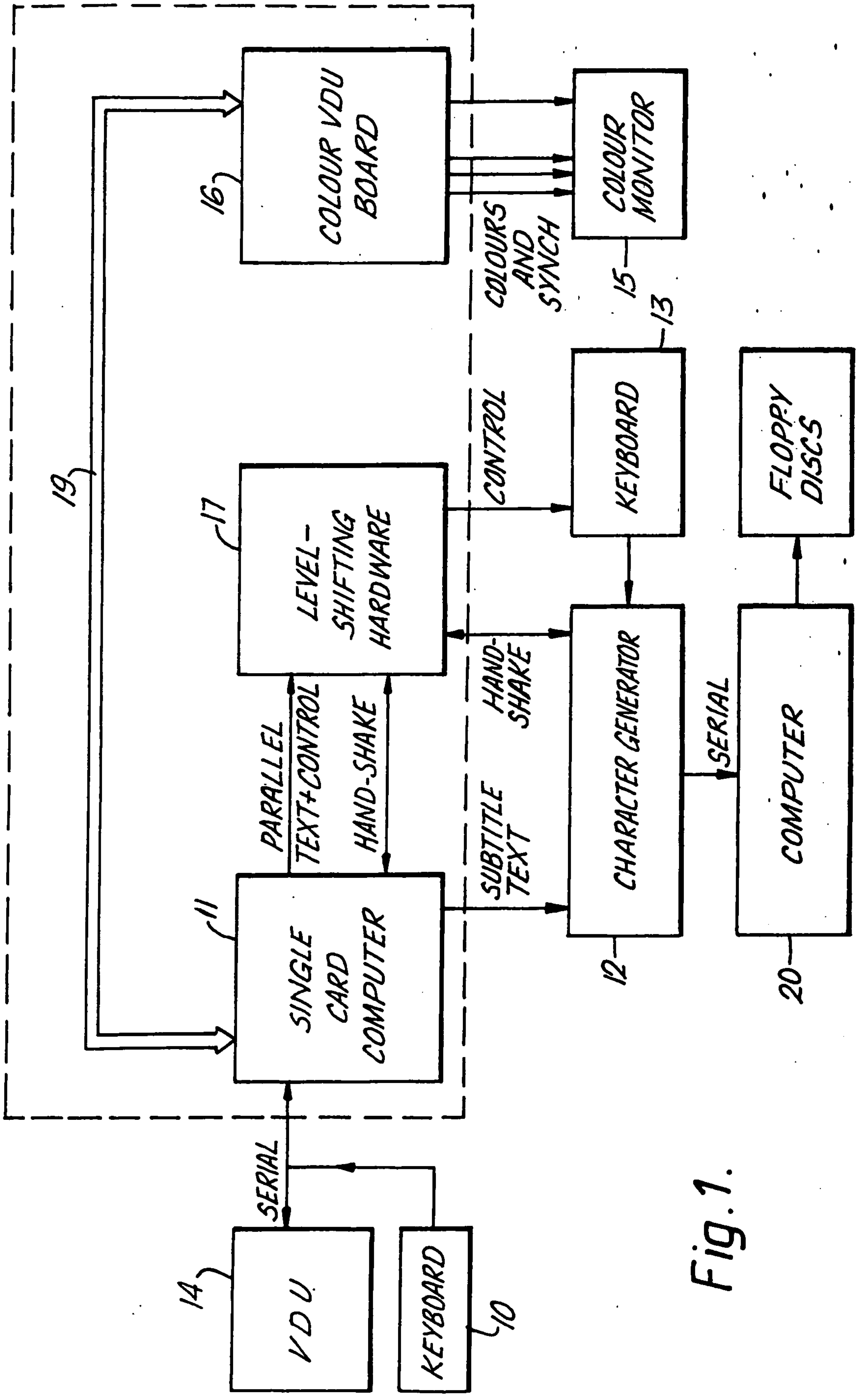
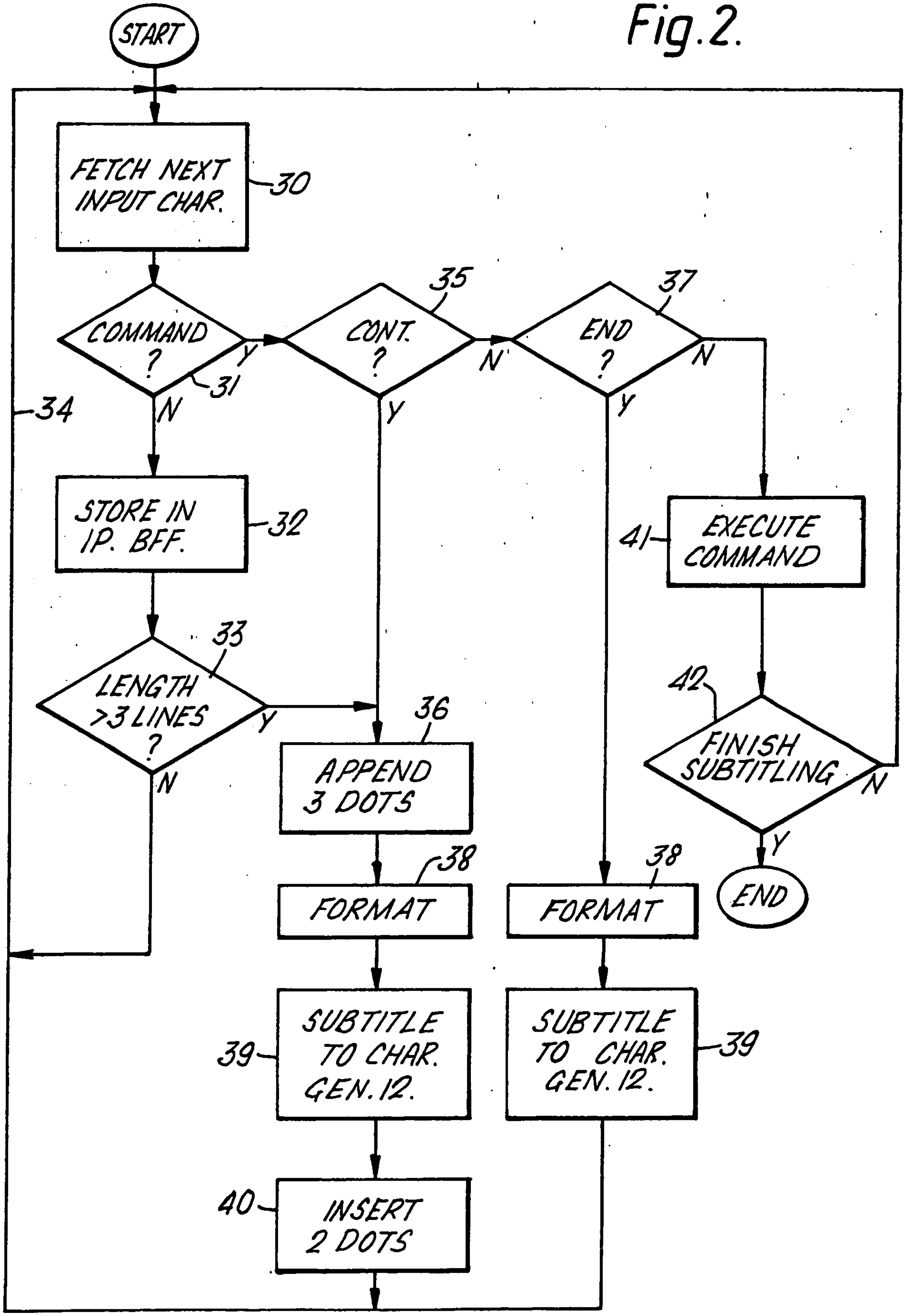

COLOUR VDU BOARD
16
COLOUR MONITOR
15
COLOURS AND SYNCH
19
LEVEL-SHIFTING HARDWARE
17
CONTROL
KEYBOARD
13
FLOPPY DISCS
HAND-SHAKE
CHARACTER GENERATOR
SERIAL
COMPUTER
20
PARALLEL TEXT+CONTROL
HAND-SHAKE
SUBTITLE TEXT
SINGLE CARD COMPUTER
11
12
SERIAL
VDU
14
KEYBOARD
10

Fig.1.

Fig.2.


START
FETCH NEXT INPUT CHAR.
30
COMMAND ?
31
Y
N
34
CONT. ?
35
N
Y
END ?
37
N
Y
STORE IN IP. BFF.
32
LENGTH >3 LINES ?
33
Y
N
APPEND 3 DOTS
36
EXECUTE COMMAND
41
FINISH SUBTITLING
42
N
Y
END
FORMAT
38
FORMAT
38
SUBTITLE TO CHAR. GEN. 12.
39
SUBTITLE TO CHAR. GEN. 12.
39
INSERT 2 DOTS
40

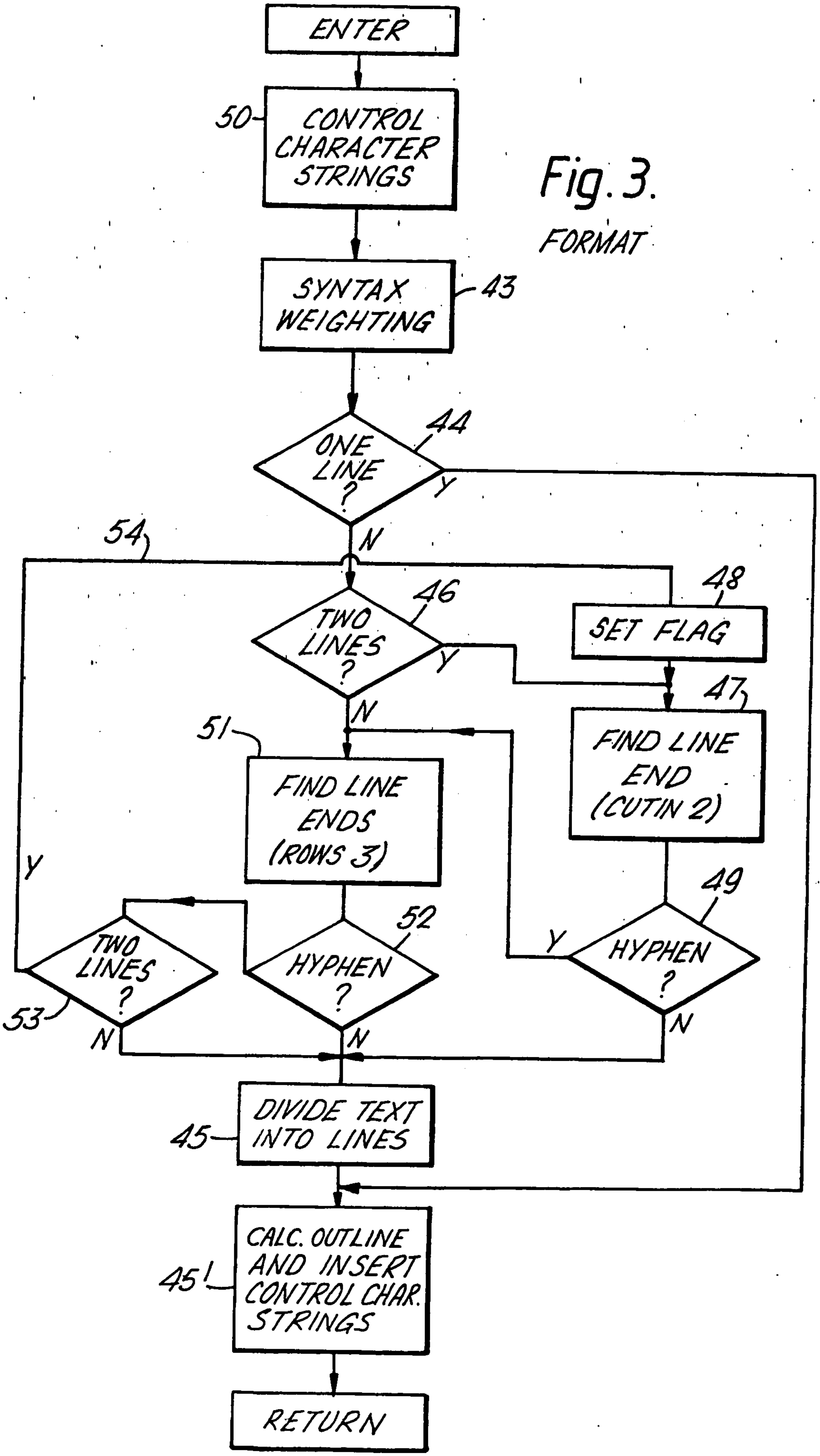

ENTER
50
CONTROL CHARACTER STRINGS
SYNTAX WEIGHTING
43
44
ONE LINE ?
Y
N
54
48
SET FLAG
46
TWO LINES ?
Y
N
47
FIND LINE END (CUTIN 2)
51
FIND LINE ENDS (ROWS 3)
Y
49
52
TWO LINES ?
HYPHEN ?
Y
HYPHEN ?
53
N
N
N
DIVIDE TEXT INTO LINES
45
CALC. OUTLINE AND INSERT CONTROL CHAR. STRINGS
45'
RETURN

Fig. 3.

FORMAT

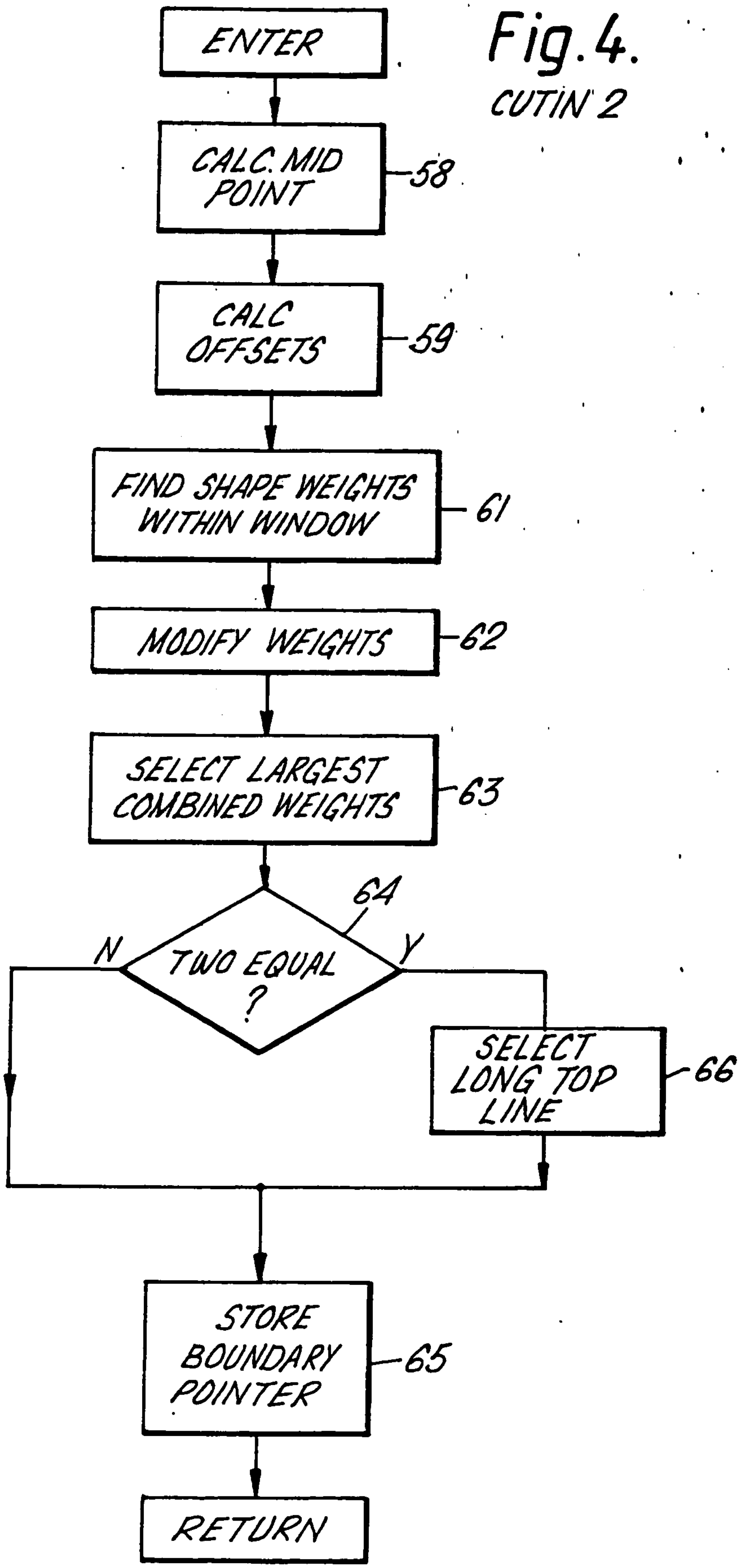

ENTER
CALC. MID POINT
58
CALC OFFSETS
59
FIND SHAPE WEIGHTS WITHIN WINDOW
61
MODIFY WEIGHTS
62
SELECT LARGEST COMBINED WEIGHTS
63
64
N
TWO EQUAL ?
Y
SELECT LONG TOP LINE
66
STORE BOUNDARY POINTER
65
RETURN

Fig. 4.

CUTIN 2

Fig. 5.

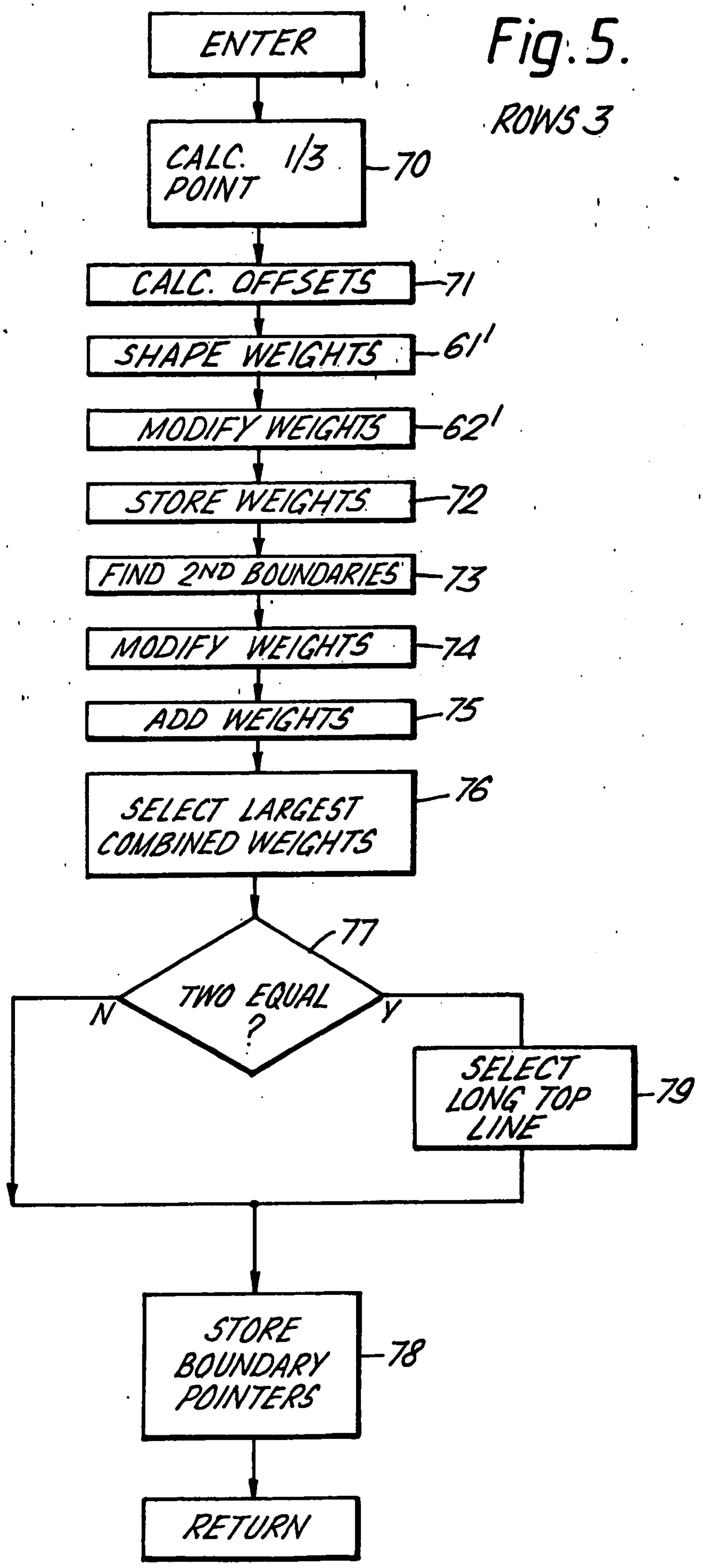
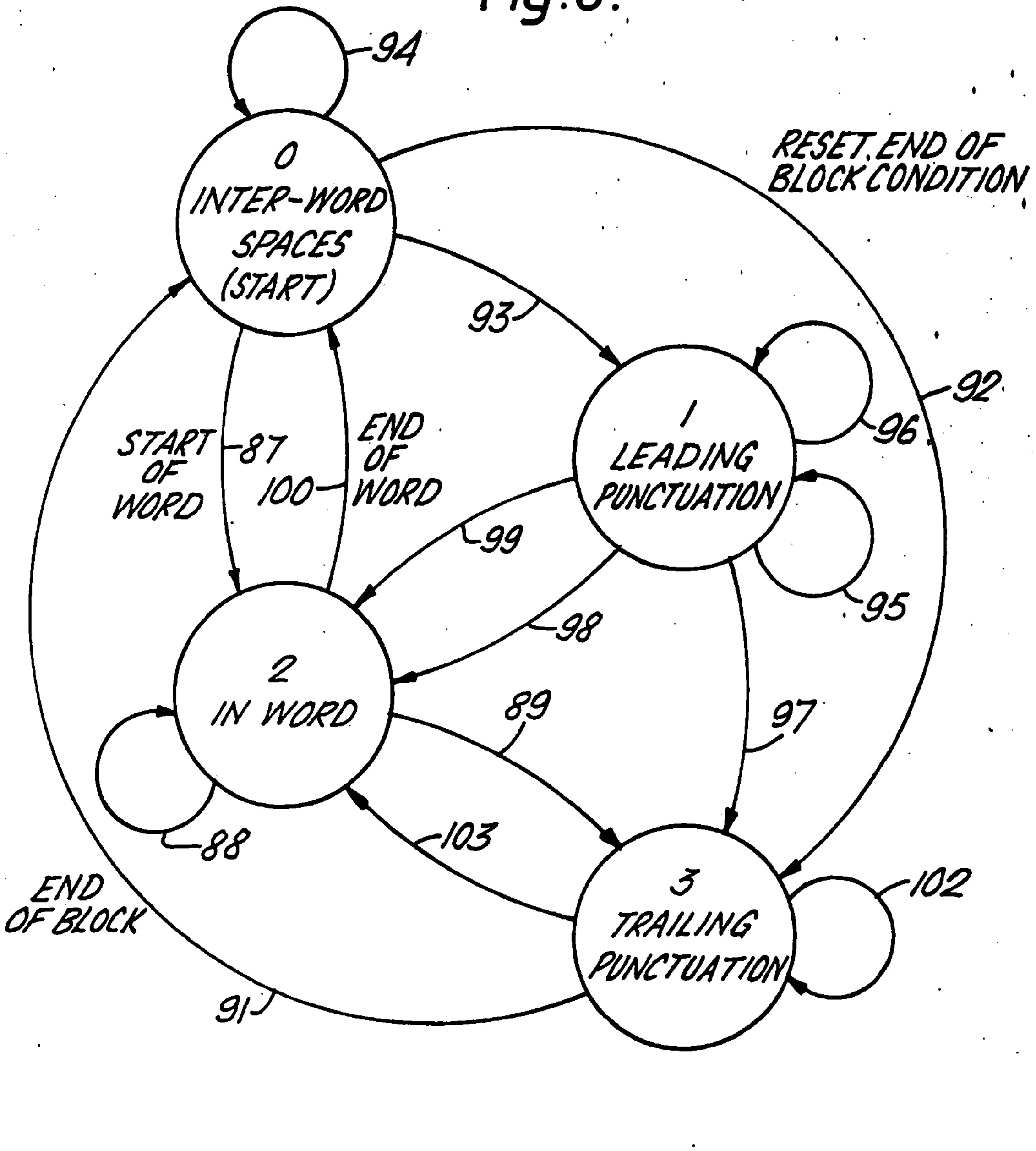

ROWS 3
ENTER
CALC. 1/3 POINT
70
CALC. OFFSETS
71
SHAPE WEIGHTS
61'
MODIFY WEIGHTS
62'
STORE WEIGHTS
72
FIND 2ND BOUNDARIES
73
MODIFY WEIGHTS
74
ADD WEIGHTS
75
SELECT LARGEST COMBINED WEIGHTS
76
TWO EQUAL ?
77
N
Y
SELECT LONG TOP LINE
79
STORE BOUNDARY POINTERS
78
RETURN

Fig.6.


94
0
INTER-WORD
SPACES
(START)
RESET END OF
BLOCK CONDITION
93
92
96
1
LEADING
PUNCTUATION
START
OF
WORD
87
END
OF
WORD
100
99
95
98
2
IN WORD
89
97
88
103
102
END
OF BLOCK
3
TRAILING
PUNCTUATION
91

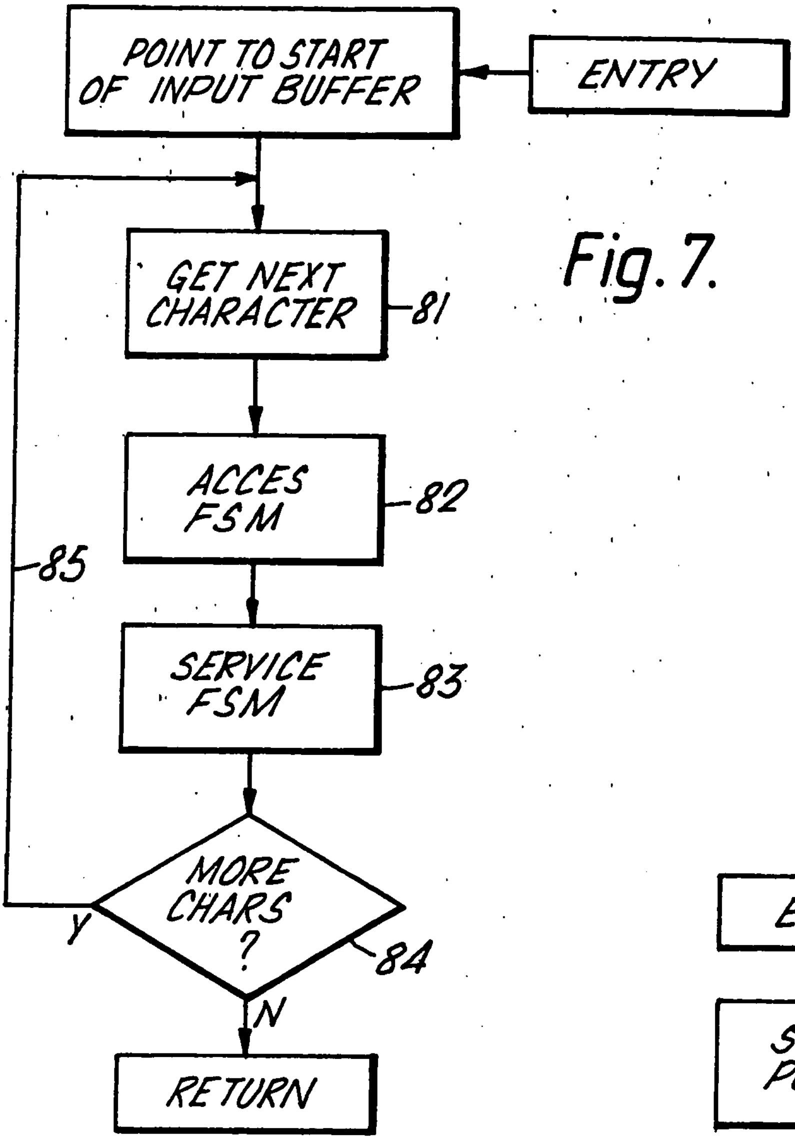

ENTRY
POINT TO START OF INPUT BUFFER
GET NEXT CHARACTER
81
ACCES FSM
82
SERVICE FSM
83
MORE CHARS ?
84
Y
N
85
RETURN

Fig. 7.

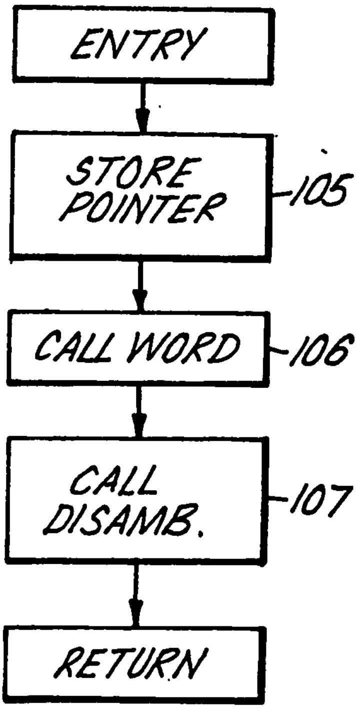

ENTRY
STORE POINTER
105
CALL WORD
106
CALL DISAMB.
107
RETURN

Fig. 8.

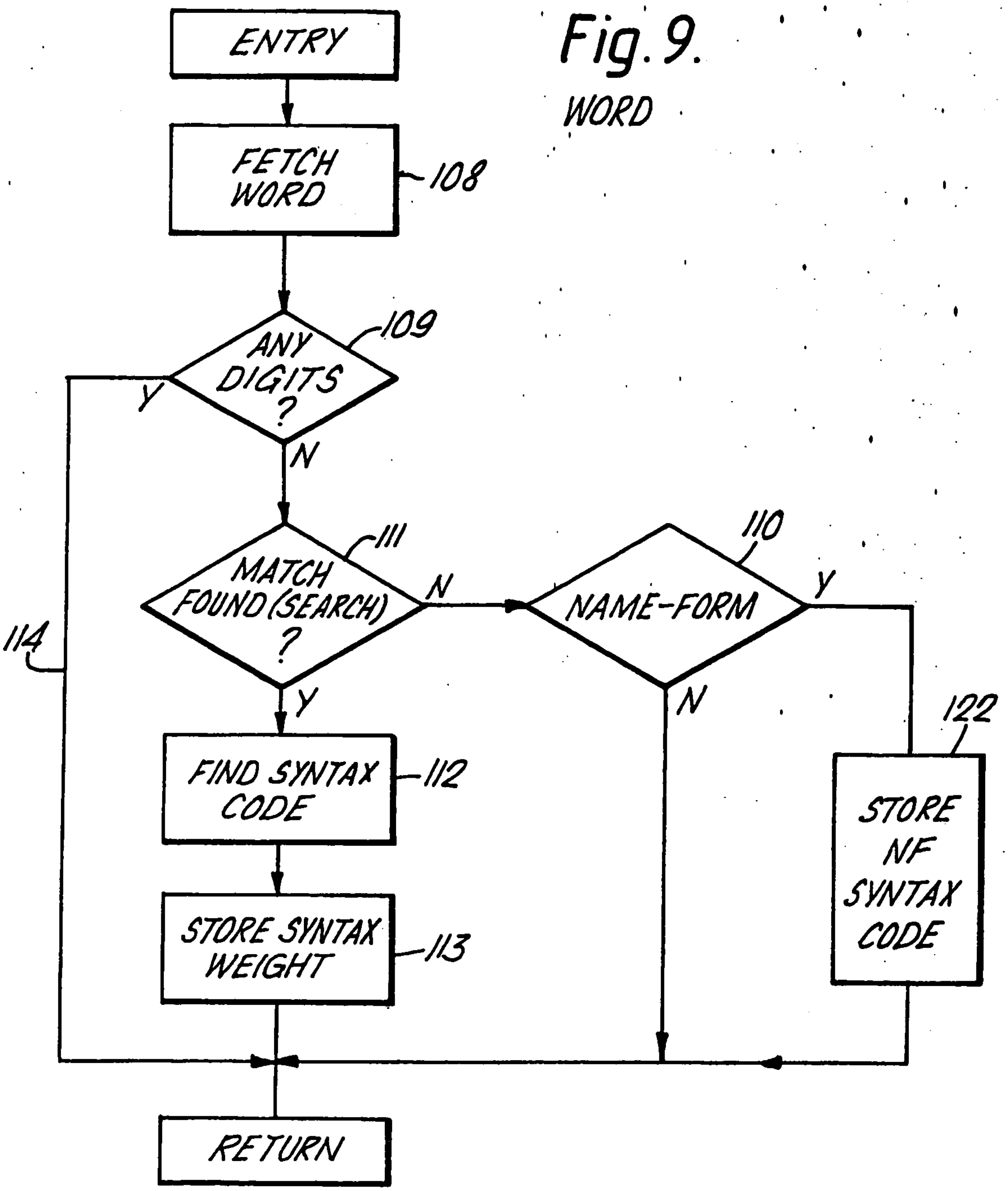

ENTRY
FETCH WORD
108
ANY DIGITS ?
109
Y
N
114
MATCH FOUND (SEARCH) ?
111
N
NAME-FORM
110
Y
N
122
STORE NF SYNTAX CODE
Y
FIND SYNTAX CODE
112
STORE SYNTAX WEIGHT
113
RETURN

Fig. 9.

WORD

Fig. 10.

SEARCH

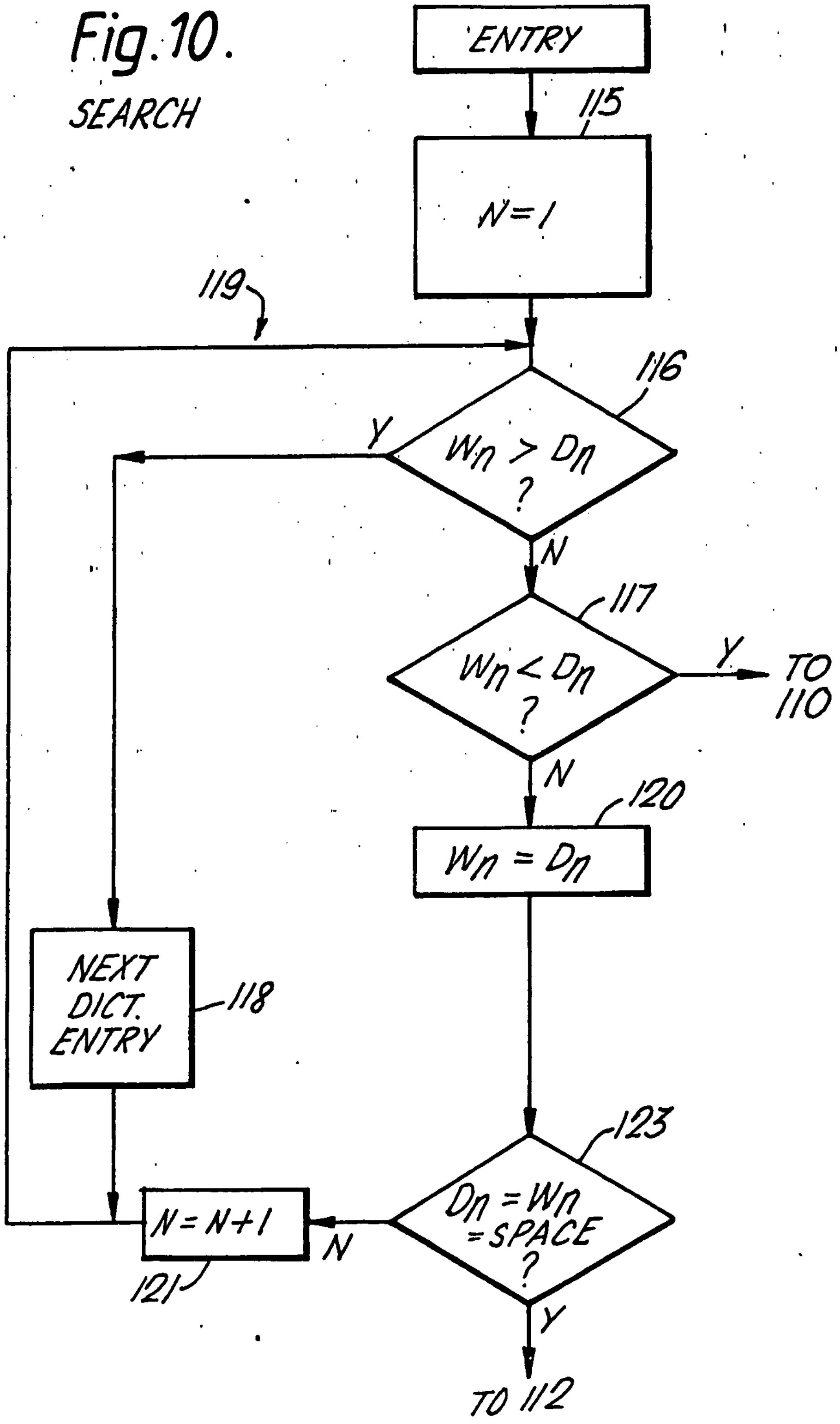


Fig. 11.

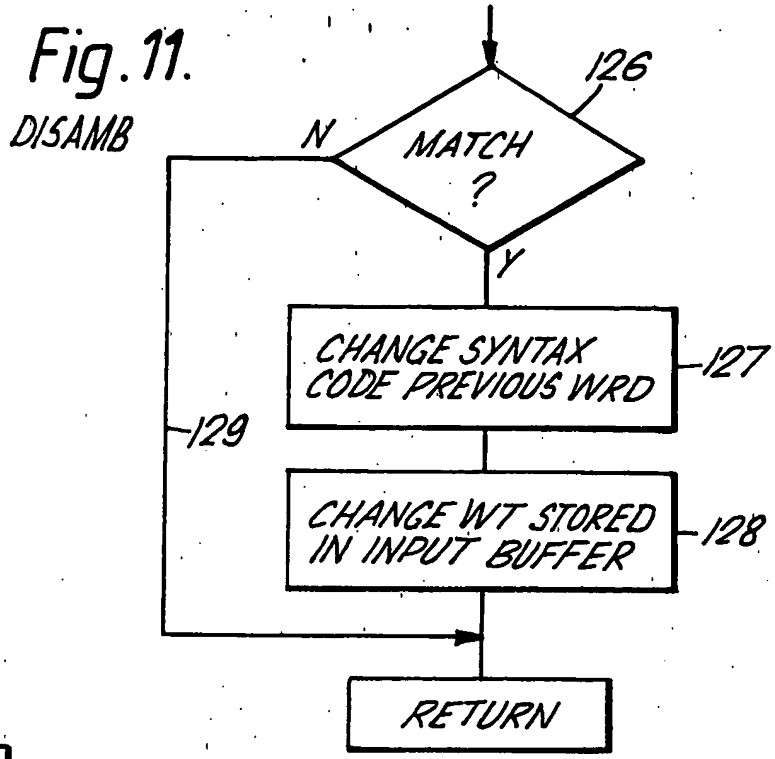
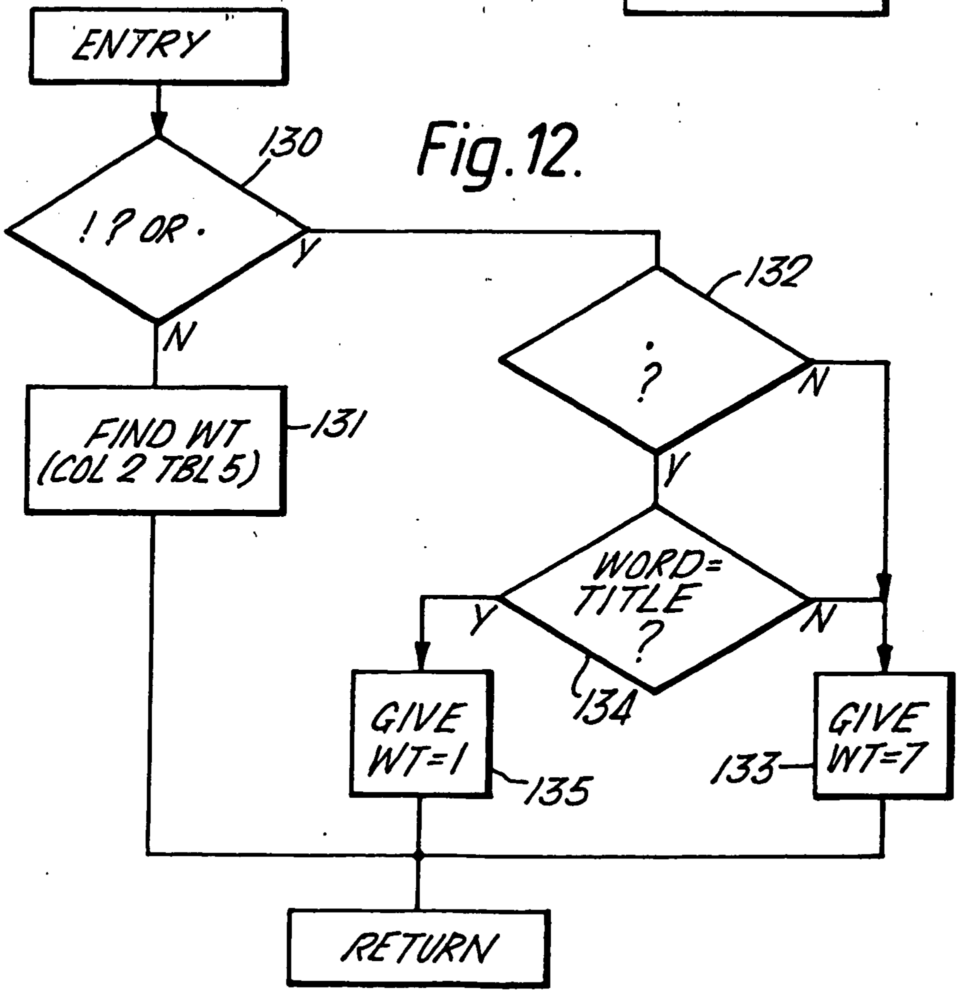

DISAMB
126
N
MATCH ?
Y
CHANGE SYNTAX CODE PREVIOUS WRD
127
CHANGE WT STORED IN INPUT BUFFER
128
129
RETURN

Fig. 12.


ENTRY
130
! ? OR .
Y
N
FIND WT (COL 2 TBL 5)
131
132
·
?
N
Y
WORD= TITLE ?
Y
N
134
GIVE WT=1
135
GIVE WT=7
133
RETURN

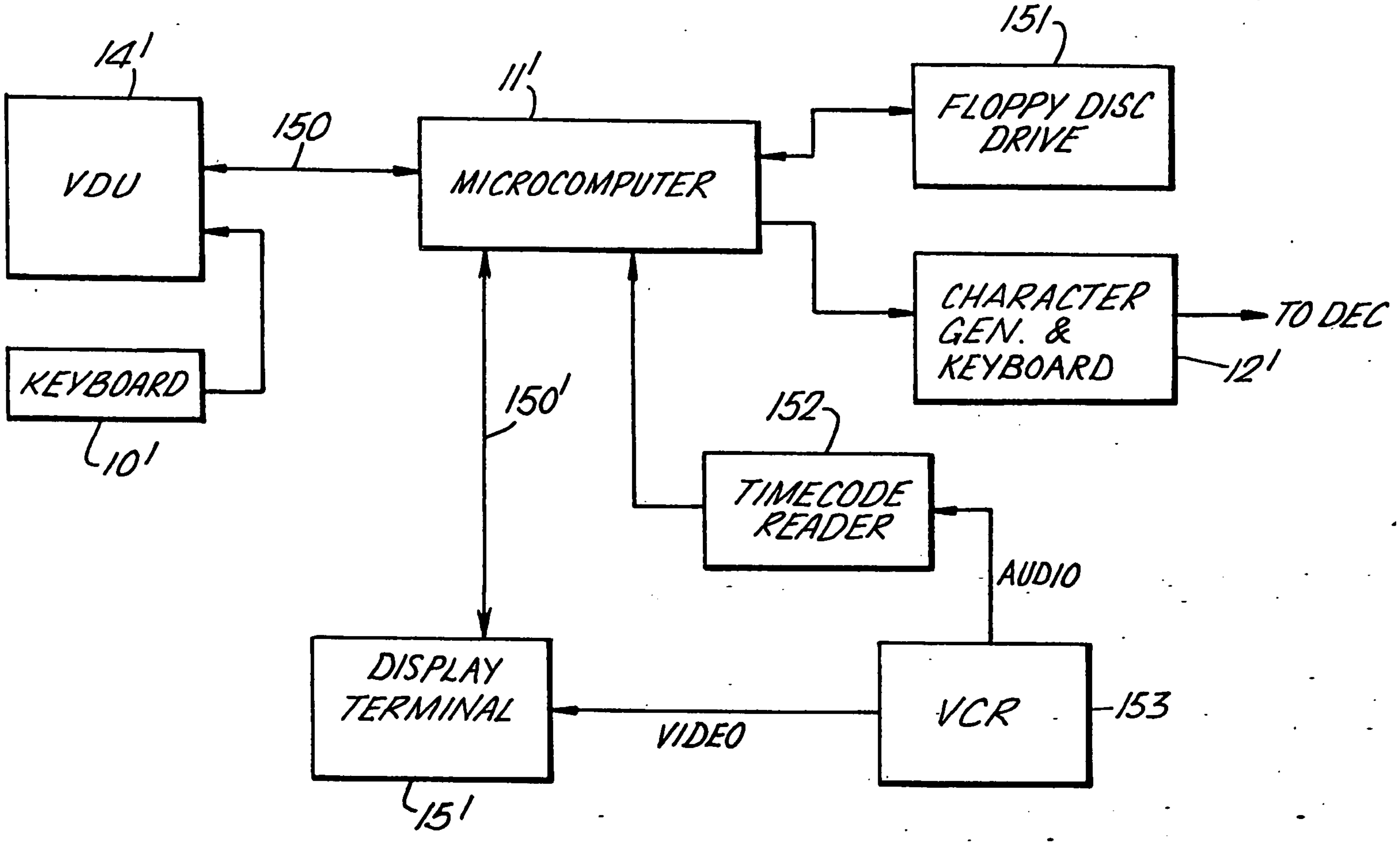

14'
VDU
150
11'
MICROCOMPUTER
151
FLOPPY DISC DRIVE
CHARACTER GEN. & KEYBOARD
TO DEC
12'
KEYBOARD
10'
150'
152
TIMECODE READER
AUDIO
DISPLAY TERMINAL
VIDEO
VCR
153
15'

Fig. 13

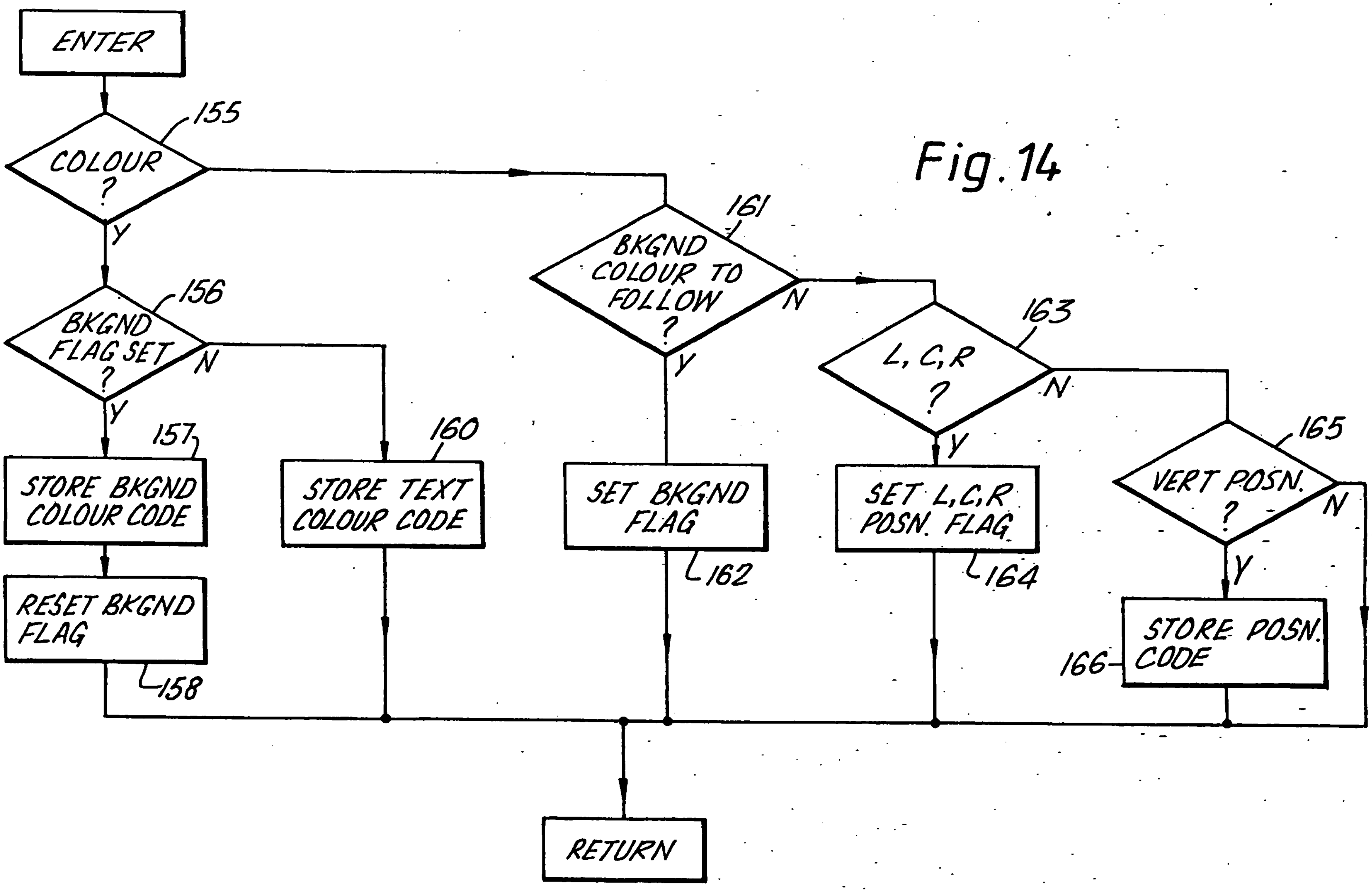

ENTER
COLOUR ?
155
Y
BKGND FLAG SET ?
156
N
Y
STORE BKGND COLOUR CODE
157
RESET BKGND FLAG
158
STORE TEXT COLOUR CODE
160
BKGND COLOUR TO FOLLOW ?
161
N
Y
SET BKGND FLAG
162
L, C, R ?
163
N
Y
SET L,C,R POSN. FLAG
164
VERT POSN. ?
165
N
Y
STORE POSN. CODE
166
RETURN

Fig. 14

SPECIFICATION

Methods and apparatus for use in arranging text

The present invention relates to methods and apparatus for use in arranging text, particularly but not exclusively for subtitling using the "Teletext" system. Subtitles of this type have particular application as an aid to the deaf who can use Teletext in order to obtain subtitles.

The Teletext system is described in the "Broadcast Teletext Specification" by the British Broadcasting Corporation, the Independent Broadcasting Authority and the British Radio Equipment Manufacturers Association, September 1976, but a brief outline now follows. A television picture is made up by interlacing two sets of 312½ lines 25 times a second but about 40 of these lines re arranged to be off the top of the picture and it is into these lines that information for the Teletext display is inserted. Currently, lines 15 - 18 during the first of each interlaced pair of pictures and 328 - 331 during the second interlaced picture of the pair are used to carry Teletext information. This information is interpreted by a decoder in the Teletext television receiver and displayed on the screen as a Teletext 'page' of graphic or alphanumeric characters in a 24-row by 40-column matrix. Each coded television data line corresponds directly to one decoded Teletext display row. The first row, or 'header', of each Teletext page contains the magazine and page number, a 16 bit code for various control functions, and 32 eight bit binary words to define the page header display. Subsequent Teletext lines for that page contain a row number and 40 eight bit words to define the characters for that row of the display. The eight bit words defining the display include control codes when that part of the line following the code is, for example, to be in a particular colour, or flashing.

By reserving a particular Teletext page as a subtitle page, subtitles may be transmitted in conjunction with a broadcast programme using the Teletext facility. A viewer with a Teletext television set is then able to select the subtitle page and superimpose the subtitles on the television picture.

Teletext subtitle pages are prepared using a computer and information for the pages is entered using a keyboard belonging to the computer. After each page has been prepared it is stored in a convenient way, usually on a floppy disc. When television transmission takes place the floppy disc is read by a computer which inserts a new subtitle page into the Teletext transmission sequence each time a subtitle is required. This procedure is synchronised by means of a timecode associated with the broadcast programme. When the continuous timecode sequence from the broadcast video tape machine matches the timecode stored with the next prepared subtitle, the subtitle is transmitted.

Existing Teletext subtitle preparation equipment is available commercially as the ASTON TCG 3 and associated keyboard, coupled with a DEC PDP computer system. Using this equipment, the person designing the subtitle must enter not only the subtitle text but also the control characters necessary to specify the subtitle display size and colour, and the boundaries of the background boxing. In addition, the subtitler must decide where the ends of subtitle lines are to occur, and must position the subtitle as required on the screen. Using such equipment is time-consuming and typically 30 hours of subtitle preparation work are required for each hour of broadcast subtitles.

In general an operator tries to achieve subtitles with line endings at convenient places which allow maximum fast comprehension by a reader and experience with television subtitling indicates that readability and perceived aesthetic "quality" of a subtitle depend on the fluency of the text, the overall shape of the subtitle and the way in which the linguistic structure is reflected in the format. Available subtitling guidelines, for example those prepared by R.G. Baker of Southampton University under the title "Guidelines for the Subtitling of Television Programmes", recommend the use of concisely structured sentences, roughly equal line lengths, if the text extends over more than one line, and the choice of line endings which correspond to major syntactic boundaries. In particular the use of hyphens is to be avoided where possible.

According to a first aspect of the present invention there is provided apparatus for use in arranging text comprising means for entering text, and analysis means for storing any text entered, analysing the text stored on the basis of linguistic criteria, and providing signals representing the text including line-end signals indicating positions for the ends of lines in arranged text, the line-end signals being derived at least partially on the basis of the said analysis.

According to a second aspect of the present invention there is provided a method for use in arranging text comprising storing text, analysing the text on the basis of linguistic criteria, and providing signals representing the text which include line-end signals indicating positions for the ends of lines in arranged text, the line-end signals being derived at least partially on the basis of the said analysis.

While the invention is expected to find many applications such as in many forms of television display or video where text is shown, in word processing and printing, particularly where the effect of respective lines on a reader is important, one of the main advantages of the invention is that the text analysis determines suitable ends for subtitle lines automatically and thus considerably reduces the time required for preparing subtitles. When an operator subtitles a programme he enters the text for a subtitle, and the apparatus automatically determines line ends. The operator then allows the subtitle produced to go to storage but the stored subtitles may later be edited. As compared with purely geometric formatting without linguistic analysis, editing is reduced by a factor of about three for two line subtitles and by about two for three line subtitles. Clearly the invention is of particular importance when subtitling is in real time for "live" broadcasts since the time available for the text input and amendment is then very limited. The subtitler merely types in the text for each subtitle, and presses a control key to initiate the automatic formatting and conversion to Teletext codes.

The apparatus and method of the first and second aspects of the invention preferebly include deriving line-end signals partially on the basis of selecting suitable lengths for a subtitle shape which allows maximum comprehension.

Apparatus according to the first aspect of the present invention preferably, in operation, inserts syntactic weights in the spaces between words in the stored subtitle text, each weight relating to the part of speech of the preceding word. Advantageously the weights entered are modified according to any trailing punctuation of the word or any punctuation preceding the next word.

As far as parts of speech are concerned some, such as adverbs, generally make suitable subtitle line ends and are therefore given high weights but others, such as the indefinite and definite articles and prepositions, make poor subtitle line ends and are given low weights. These weights are given in column 1 of Table 5 below.

Similarly some punctuation, such as a fullstop or a question mark, usually makes suitable line endings and therefore punctuation is used to modify weights as shown in column 2 of Table 5.

As far as shape is concerned the most suitable line end for a two line subtitle occurs in the middle of the text and suitability decreases towards both ends of the text. The values in Table 1 given below provide a multiplying factor for each word boundary on the basis of its position along the subtitle. The multiplying factor is used to modify the above mentioned syntactic weights to produce a combined boundary weight.

An embodiment of the invention will now be described by way of example, with reference to the accompanying drawings, in which:-

Figure 1 is a block diagram of apparatus according to the invention,

Figure 2 is an overall flow diagram employed by a computer 11 of Figure 1.

Figure 3 is a flow diagram of a subroutine FORMAT of Figure 2,

Figure 4 is a flow diagram of a subroutine CUTIN 2 of Figure 3,

Figure 5 is a flow diagram of the subroutine ROWS 3 of Figure 4,

Figure 6 is a chart of a finite state machine operated by the computer 11 of Figure 1,

Figure 7 is an overall flow chart for operating the finite state machine of Figure 6,

Figure 8 is a flow chart of a Task 0 of the finite state machine,

Figure 9 is a flow diagram of a subroutine WORD of Figure 8,

Figure 10 is a flow diagram of subroutine SEARCH of Figure 9,

Figure 11 is a flow diagram of a subroutine DISAMB of Figure 8,

Figure 12 is a flow chart of a Task 2 of the finite state machine,

Figure 13 is a block diagram of alternative apparatus according to the invention, and

Figure 14 is a flow diagram showing how various control parameters are set.

In Figure 1 subtitles are entered using a keyboard 10 and passed to a "Cromeco" single card computer 11 which is constructed and programmed to process subtitled text, this text being passed to an ASTON TCG-3 character generator 12 and thence to a DEC PDP subtitling computer 20. In operation of the unmodified ASTON and DEC equipment a Teletext page editing keyboard 13 is used to enter subtitled text and standard function keys are used for the manual formatting, boxing, colouring and positioning of the subtitles. When using this embodiment of the invention to prepare subtitles, a VDU 14 and a keyboard 10 are used to operate the computer 11. A colour monitor 15 interfaced by a "Hi-Tech" S-100 colour VDU board 16 shows the most recent subtitle formatted by the computer 11 in the form of a Teletext page. Level shifting hardware 17 is required to provide handshake signals in the required form for the character generator 12 in order to allow the subtitle pages to be transferred via the character generator 12 to the computer 20. Since some control signals for the computer 20 are normally generated by the Teletext keyboard 13, the hardware 17 also provides these signals which are passed by way of the keyboard. A standard S-100 bus 19 couples the computer 11 to the board 16.

An alternative apparatus for entering subtitles is shown in Figure 13, where subtitles are entered on a keyboard 10' connected to a VDU 14' which may be type CIT-101 manufactured by C-ITOH. The VDU 14' is connected by way of an RS-232 serial interface represented by a connection 150 to a microcomputer 11'. A suitable microcomputer for this purpose is based on a Comart communicator containing an S-100 bus and the following cards:-

a Z80 CPU card,
a RAM card,
a ROM card,
a floppy disc control card (16 FDC) to control a twin floppy disc drive 151, and
a time code port for a time code reader 152.

An operator views a program to be subtitled on the screen of a terminal 15' which may, for example, be a D.M. England "Miracle" terminal. Video and audio for the terminal 15' are provided by a video cassette recorder (VCR) 153 and since one of the functions of the terminal 15' is to combine subtitle text with the television picture to allow subtitles which have already been recorded on floppy disc to be reviewed, the terminal 15' is connected by way of an RS-232 interface 150' to the microcomputer 11'. The terminal 15' is arranged to switch between the video signal from the VCR 153 and subtitle information provided by way of the microcomputer 11' in order to produce a display of subtitles superimposed on the picture.

On each video tape a second audio channel carries a serial data time code which may for example originate from a studio master clock when the program is recorded. This time code is used to control the time at which subtitles are displayed and for this purpose time codes appearing on the floppy discs containing subtitles are compared with time codes from the program recording. When a match occurs the terminal 15′ displays a subtitle. The time code reader 152 reads the code in the second audio channel and supplies it to the time code port of the microcomputer 11′ but since the time code reader provides a 32 bit parallel signal (8 bits each for hours, minutes, seconds and frames) and the S-100 bus can only receive 8 parallel bits, the time code port selects 8 bits at a time when the port is enabled and applies them to the bus in four groups of 8 bits. A suitable time code reader is the Avitel type 2030.

In general in Figure 13 the various codes used for subtitling can be recorded in any convenient form using the floppy disc drives 151. A special data protocol is required in Teletext subtitling to transfer subtitled data to a DEC PDP subtitling computer and for this reason the S-100 bus is connected to an ASTON TCG-3 character generator (and keyboard) 12′ via level shifting hardware similar to that shown at 17 in Figure 1 which provides the required protocol for the DEC computer. In this way subtitles produced can be recorded either directly on floppy discs or passed to a DEC subtitling computer.

In an alternative arrangement the terminal 15′ may be replaced by a colour monitor which receives audio directly from the VCR 153 but receives video by way of video switcher and card connected to the S-100 bus in the microcomputer 11′. A HI-TECH VDU 5 card provides a video signal containing the subtitle text, and the video switcher switches from picture to subtitle information in order to superimpose the text on the picture.

Since both Figures 1 and 13 employ commercially available equipments which can be interfaced using the manufacturer's data no further details of interconnections are given. There now follows a series of flow charts and accompanying description which will allow either computer 11 or 11′ to be programmed.

The operation of the apparatus of Figure 1 and in particular the computer 11 is now described by means of flow diagrams.

In preparing subtitles for later transmission, as a subtitle is entered on the keyboard 10 or 10′ the computer 11 or 11′ acquires each new input character in an operation 30 of Figure 2. The character is tested in an operation 31 to discover whether it is a command or a text character. If it is not a command it is stored in an input buffer of the computer 11 or 11′ in operation 32 and the buffer length is tested in a test 33 to determine whether the text so far stored in the input buffer is longer than three maximum length subtitle lines. If not the next input character is acquired by means of a loop 34. Should the test 31 indicate that the current character is a command then a test 35 determines whether the command is for the insertion of a continuation. If so three dots are appended to the current text buffer in an operation 36, and a subroutine FORMAT, described below, is called in an operation 38 to format the subtitle into a Teletext page. Then, in an operation 39 the page is sent to the character generator 12, the input buffer is cleared, and an operation 40 is carried out to place two continuation dots at the start of the cleared buffer. The operations 36 to 40 are also carried out if the test 33 indicates that a subtitle held in the input buffer is longer than three lines. If the test 35 shows that the command is not for a continuation, a test 37 is carried out to determine whether the command indicates the end of the subtitle, specified by the operation of an 'end of subtitle' key. If so the subroutine FORMAT is called (operation 38) to format the subtitle into a Teletext page and is followed by the operation 39 and a jump back to the operation 30. If the test 37 indicates that the command is another possible command, such as edit, clear, colour, position, control, this command is executed in an operation 41 by the computer 11 and then, unless the command indicates the end of subtitling in a test 42, the next input character is acquired by means of a jump 42 back to the operation 30.

In order to control colour and position the following parameters have to be set:-

Character height (always double height for subtitles in Teletext),
Text colour (7 possible colours),
Background colour (8 possible colours),
Vertical position (which can be in the top, middle or bottom eight lines and within those areas the bottom line of the subtitle can be positioned up one, two or three double height lines), and
Horizontal position (left, centre or right as far as this is possible allowing for control characters).

Except for the vertical position, these parameters are recorded as part of the subtitle in that the control characters necessary to produce the required display are in general inserted before or after the text of each subtitle line. The required vertical position is achieved by specifying in which lines of a page the subtitle is to appear. Some of the characters used are now listed:-

DH setting the characters in the subtitle to double height,
SB/EB to position starting the box around a subtitle and ending it - two such characters are required in order to ensure that spurious signals do not start or end a box prematurely,
TC text colour,
BC background colour, and
NB to indicate the start of a new background colour.

Control characters as listed above are inserted into the text near the end of the FORMAT subroutine.

As part of operation 41 in Figure 2 the above parameters are set using the flow diagram of Figure 14. Special keys on the keyboard 10 or 10′ are used to enter commands and a test 155 is first carried out to determine whether the command entered is a colour command. If so a test 156 determines whether a background flag has been set (since setting subtitle background colour is carried out by depressing a first key to indicate background and then a colour key) and if it has an operation 157 stores the background colour. The background flag is then reset to zero in an operation 158 and a return to operation 42 of Figure 2 occurs.

If the test 156 indicates that the background flag is not set then the colour command relates to text colour and therefore text colour is stored in an operation 160 before return to the operation 42.

If the test 155 indicates that the command does not relate to colour then a test 161 determines whether the command indicates that a background colour is about to be set when an operation 162 sets the background flag before return. If the test 161 indicates that the test is not related to background colour then a test 163 is carried out to determine whether the command relates to the horizontal position of the subtitle which can be specified as left, centre or right (LCR) and if so L, C or R flags are set in an operation 164. If the test 163 indicates that the command is not related to horizontal position then a test 165 is carried out to see if the command relates to vertical position. If not there is a return to operation 42 but otherwise a position code relating to vertical position is stored in an operation 166.

In the FORMAT subroutine of Figure 3 the control characters stored in the operation of Figure 14 are assembled into a first string to be inserted before the text of each subtitle line in a later operation, and a second string to be inserted after the text of each subtitle line. The first string is in the form DH, (BC), (NB), TC, SB, SB where control characters in brackets are optional, and the second string is in the form (BB) (EB) (EB) where again the characters in brackets are optional in the sense that if there is no room for them in a line they are not required since the line continues to the right hand end of the display. The assembly of the two strings and subsequent storing for use later is carried out in an operation 50. A subroutine "SYNTAX" is next called in operation 43 to analyse the linguistic content of the subtitle and insert syntax weights in the spaces between each word. SYNTAX is described in more detail below. Following operation 43 a test 44 is carried out to decide whether the number of input characters in the input buffer will fit on one subtitle line allowing for the mandatory control characters previously specified. If so the whole subtitle will fall on one line and it is output to the computer 20 via the character generator 12 in operations 45 and 45′ described in more detail below. However if the test 44 indicates that there is more text than can be presented in one line, a test 46 is carried out to discover whether the text will fit on two lines. If so a subroutine "CUTIN 2" is called in an operation 47 to determine the half-way point in the text and to determine a "window" in which a suitable point for the end of the subtitle line occurs.

The need for a "window" arises in the following way: it is a primary object in subtitling to avoid the use of hyphens if possible and if a subtitle takes up nearly the whole of two lines then when it is broken at a point between words it may be that this point does not allow each resulting part of the text to fit on one line. The window ensures that any space chosen for the end of the line will allow the two resulting parts of text to fit on respective lines. The subroutine "CUTIN 2" then modifies the syntax weights, as is described below, and selects the word end, if any, with the highest modified weight within the window. A test 49 is then carried out to determine whether a word end occurs in the window since if not then a two line subtitle would require a hyphen. If a hyphen would be required it is better to put the subtitle on three lines and to this end a subroutine "ROWS 3" is called in an operation 51, described below and relating to three lines. If a hyphen is not required the operations 45 and 45′ are carried out to divide the text into two lines at the point indicated by the operation 47, and carry out other tasks described below.

Should the test 46 indicate that the text will not fit on two lines then the operation 51 is carried out in which first a point one-third of the way from the beginning of the subtitle is found by calling ROWS 3. As will be described below ROWS 3 uses CUTIN 2 to find, for each word boundary in the window, a corresponding window for the end of the second line and a corresponding best line end for the second line. In this way pairs of possible first and second line ends are generated and ROWS 3 chooses the best such pair. A test 52 is then carried out to determine whether the operation 51 has been unable to find a pair of line ends at ends of words and therefore whether a hyphen is required at the end of one or both of the lines. If so a test 53 determines whether the subtitle would fit on two lines and if so then a loop 54 back to the operation 47 occurs but a flag is also set (operation 48) to ensure that when the test 49 is carried out a jump to the operation 45 always occurs. If the result of the test 53 indicates that the text will not fit on two lines then a hyphen must be used and the operations 45 and 45′ are carried out to divide the text into three lines, box the subtitle so prepared and pass it to the character generator 12.

In deciding how much text can be included in a line an allowance must be made for the first string of control characters. The second string can be ignored if there is not room for them in a line since they close the box and reset the background colour and this is unnecessary if the text extends to the end of the line.

The first operation CUTIN 2 (Figure 4) is an operation 58 in which the mid-point of the current subtitle is calculated. In order to define the window surrounding this mid-point a lower offset C from the mid-point and an upper offset B are calculated in an operation 59 from the following:-

$C = M - (N-N')$, and

$B = M - (N'+1)$,

where M = the maximum number of characters per Teletext display line, less the maximum number of control characters in the first string,

N = the number of characters in the subtitle, and

N′ = half the number of characters in the subtitle.

However, in order to ensure that the window does not extend beyond the limits of the input buffer, if $C>N'-1$ then C is set to $N'-1$ and similarly if $B>N-N'$ then B is also set to $N-N'$.

As has been explained the shape of a subtitle is also important in determining the readability of the title. For this reason the weights determined by syntax weighting are multiplied by factors relating to shape. The computer holds Table 1 giving factors representing the suitability of positions (in terms of subtitle length) for ends of subtitles.

TABLE 1

WORD BOUNDARY LOCATION	SHAPE WEIGHTING FACTOR:	
(% OF SUBTITLE LENGTH)	TWO LINE	THREE LINE
0 - 3	0	0
4 - 6	0	0
7 - 9	0.25	0.25
10 - 12	0.25	0.25
13 - 15	0.25	0.5
16 - 18	0.25	0.5
19 - 21	0.5	0.75
22 - 24	0.5	0.75
25 - 27	0.5	1.0
28 - 30	0.5	1.0
31 - 33	0.75	0.75
34 - 36	0.75	0.75
37 - 39	0.75	0.5
40 - 42	0.75	0.5
43 - 45	1.0	0.25
46 - 48	1.0	0.25
49 - 51	1.0	0
52 - 54	1.0	0
55 - 57	0.75	0.25
58 - 60	0.75	0.25
61 - 63	0.75	0.5
64 - 66	0.75	0.5
67 - 69	0.5	0.75
70 - 72	0.5	0.75
73 - 75	0.5	1.0
76 - 78	0.5	1.0
79 - 81	0.25	0.75
82 - 84	0.25	0.75
85 - 87	0.25	0.5
88 - 90	0.25	0.5
91 - 93	0	0.25
94 - 96	0	0.25
97 - 100	0	0

For two line subtitles the shape factor of column 2 is used and peaks at the centre of the subtitle (for three line subtitles (column 3) there are two peaks at positions one-third and two-thirds of the subtitle length).

In operation 61 the appropriate multiplying factors for each possible subtitle line-end at a word end, if any, in the window are obtained from column 2 of Table 1, and in operation 62 these weights are multiplied by the syntax weights already inserted by the subroutine FORMAT in the input buffer in the spaces between words of the subtitle. The products so found are stored for use in an operation 63 which selects the largest weight or the two largest weights. If a test 64 shows that no two weights are equal an operation 65 sets a pointer to the line-end corresponding to the largest weight. If there are two equal largest weights then in an operation 66 a pointer is set which points to that boundary which will give a top line which is longer than the next line.

Where a subtitle is to consist of three lines the operation 51 of Figure 3 calls the subroutine ROWS 3 shown in Figure 5. In an operation 70 the point one-third of the way along the subtitle is calculated and in operation 71 upper and lower offsets or a window around this point are calculated as follows:-

B = N−N″+1
C = 2M−(N−N″)
where N″ = N/3.

In order to ensure that the window does not extend beyond the beginning and end of the text buffer B is set to N″ if
B>N−N″, and
C is set to N″ if
C>N″−1.

Using Table 1 column 2 shape factors for each possible line-end at a word end, if any, in the window are found in an operation 61′ and the syntax weights in the spaces between words corresponding to these boundaries are multiplied by the corresponding shape factors in an operation 62′. The weights so found are stored in an operation 72 and then for each possible line-end in the window found in operation 71, a window is found in an operation 73 for a second line-end using the techniques described in Figure 4 and subroutine CUTIN 2. For each of these possible line ends, the syntax weights stored by the subroutine FORMAT are multiplied by appropriate factors from column 2 in Table 1 (operation 74). In an operation 75 the respective pairs of combined weights corresponding to pairs of possible line-ends are added together and the largest combined pair is determined in an operation 76 in order to find the most satisfactory pair of boundaries. A test 77 determines whether two pairs of combined weights are equal and if not an operation 78 stores pointers to the two line-ends. If there are two equal combined pairs then the ~~pair~~ giving the longest top line are selected in an operation 79 and pointers to these line-ends are stored.

More details of the operations 45 and 45′ of Figure 3 are now given. In the operation 45 each line is copied into an output buffer for storing eight rows of forty characters corresponding to a third of the page (that is either the top, centre or bottom of the display). Each line is stored separately and the copying is carried out on the basis of the pointers to the ends of lines already found. In the operation 45′, the longest line is found and the subtitle is positioned to the left, centre or right of the picture as represented by position in the output buffer according to a previous keyboard command stored in the operation 164 of Figure 14. Positioning must allow for insertion of the first string of control characters at the left hand end of the first line. Outut buffer position also depends on the vertical position code stored in the operation 166 of Figure 14. The first and second strings of Teletext control characters are inserted around the subtitle text to produce a regularly boxed, double-height display of the required colour. The operation 45′ may include a comparison of the length of the longest line with the other line or lines. If the latter is less than four spaces shorter than the former, the second string is inserted at the end of the text, otherwise the second string is inserted in a horizontal position corresponding to the string stored with the longest line. In this way the ends of the box are aligned vertically unless the lines differ in length by more than four characters, when the box is tailored to fit the shape of the text.

If completed subtitles are to be stored on a floppy disc the contents of the output buffer, representing eight lines of a Teletext page are output to disc. If subtitles are to be stored in a form suitable for the DEC computer a complete Teletext subtitle page is passed by way of the character generator 12 to the computer 20 where it is stored on floppy disc or entered directly into the Teletext system. In this case one of the computers 11 and 11′ inserts the other 16 blank lines before the page is transferred to the character generator 12. For real time subtitling the computer 20 inserts the page prepared into the Teletext "magazine" immediately.

One of the most important parts of the present invention will now be described in more detail and that is the operation 43 of Figure 3 in which syntax weights are inserted in the spaces between words. For this operation a finite state "machine" (FSM) shown in Figure 6 is serviced according to the flow diagram of Figure 7.

The FSM can exist in one of four states 0, 1, 2 and 3, and each time a character in the input buffer is examined in an operation 81 the FSM treats this character as an input and follows one of the arrows shown in Figure 5 in accordance with its present state and the input character. Each arrow determines a route between states (or back to the same state) and usually includes one or two tasks. Thus the finite state machine is accessed in an operation 82 and the tasks are serviced in an operation 83 so that at the end of operation 83 the machine is usually in a new state. Then if there are more characters, as is determined by a test 84, the machine is accessed again by means of a loop 85 and again moved, usually, to a new state.

The operation of the finite state machine is explained with reference to Table 2. Note that in this table * represents characters such as *, $, £, %, &

TABLE 2

FROM STATE TO STATE	DESIG.	CHARACTERS	OBJECTS	TASK NOS.
0-0	94	Space	Ignore extra spaces	
0-1	93	."-'(	Record punctuation weight	2
0-2	87	A-Z a-z 0-9*	Include punctuation weight in weight stored Reset for new word	3 4
0-3	92	)?!:;,	Reset boundary Record punctuation weight	1 2
1-1	95	Space	Ignore extra spaces	
1-1	96	."-'(	Record punctuation weight	2
1-2	98	A-Z a-z *	Include punctuation weight in weight stored Reset for new word	3 4
1-2	99	0-9	Include punctuation weight in weight stored Reset	3 4
1-3	97	)?!:;,	Record punctuation weight	2
2-0	100	Space	End of word. Find syntax class Store syntax weight	0 5
2-2	88	* A-Z a-z)	Traverse word	
2-3	89	):-,;"'?!.	End of word. Find syntax class Record punctuation weight	0 2
3-0	91	Space	End of word. Find syntax class	5
3-2	103	* A-Z a-z 0-9 (	Reset punctuation weight	6
3-3	102	:-,;"'?!.)	Record punctuation weight	2

If the machine is in a certain state (the first figure in column 1) then the character determined in the operation 81 will fall into one of the classes given in column 3 and this determines the route and state (second figure in column 1) to which the machine moves in operations 82 and 83. Column 4 explains the object of the task or tasks, if any, to be carried out in traversing the route and column 5 gives the number of the task and hence allows the appropriate subroutine to be followed to carry out that task. For example supposing the machine is in the state 0 between words and the first character of the next word is A. Since A is given in column 3 for the state 0 the machine then moves to state 2 via the route designated 87 carrying out tasks 3 and 4 as given in column 5. Task 3 is to include the punctuation weight in the current space between words and task 4 is to reset the machine for the next word. Possible subtitle line-ends are referred to as boundaries in describing the FSM and these boundaries occur at the ends of words unless there is trailing punctuation in which case the boundary occurs in the first space after the trailing punctuation. Weights are stored in the input buffer at boundaries.

Usually after the start of a word a number of lower case letters will occur so that starting in state 2 with a lower case letter the route 88 will be traversed back to state 2 for each letter. Assuming that a word ends with a space a route 100 from state 2 to state 0 will be followed in which tasks 0 and 5 are carried out to find the syntax code for the word and store this weight in the input buffer at the end of the current word. Should the last letter of a word be followed by punctuation such as a fullstop, a route 89 will be followed instead of the route 100 and a task 2 will be carried out instead of the task 5. Task 2 determines the punctuation type and finds the effect on the syntax code of this punctuation. If a space now occurs a route 91 is followed from state 3 to state 0 in which the task 5 is carried out.

Trailing punctuation can sometimes follow a space and then after the route 100, a route 92 will be followed in which the task 2 is followed by a task 6 in which the weight stored is re-set to allow for the punctuation now found.

Rather than giving a lengthy description containing observations about each route between states, the complete operation of the FSM can best be understood from the designations marked on Figure 6, Table 2 and the following list of tasks:-

TASK 0: "END OF WORD"
- match word with syntactic dictionary
- find syntax code if matched
- disambiguate previous word and boundary if possible

TASK 1: "RESET BOUNDARY"
- restore prematurely stored boundary to space

TASK 2: "FIND COMBINED BOUNDARY WEIGHT"
- determine punctuation type
- find the combined punctuation/syntax weight using column 2 of Table 5
- retain the combined weight for later use

TASK 3: "STORE COMBINED WEIGHT"
- store combined punctuation/syntax weight at boundary location of latest word

TASK 4: "RESET FOR NEW WORD"
- keep copy of previous syntax code
- reset variables for new word

TASK 5: "STORE SYNTAX WEIGHT"
- store syntax weight in place of the first space following a word and its trailing punctuation

TASK 6: "RESET PUNCTUATION WEIGHT"
- reset if punctuation incorrectly identified.

The way in which each of the tasks is carried out will now be described.

The overall flow diagram for Task 0 is shown in Figure 8 and comprises an operation 105 in which the position of the end of a word in the input buffer is stored by means of a pointer. Next a subroutine WORD is called in an operation 106 and accesses a dictionary where words are filed against syntax codes. Since some of these codes are ambiguous another subroutine is called in operation 107 to disambiguate the previous word using the current word if this is possible.

In the subroutine WORD which is shown in Figure 9 the current word is first copied from the input buffer into a word buffer and then a test 109 is carried out to determine whether there are any digits in the word. If not then a subroutine SEARCH is called in a test 111 to search the syntax dictionary for a matching word. If a match is found a syntax code is allotted (in an operation 112) which describes the part of speech of the word. This code is then stored in an operation 113. Should tests 109 detect the presence of digits a branch 114 to the end of the WORD subroutine is taken. If a dictionary match is not found in the test 111, the WORD subroutine performs a test 110 for the occurrence of a "proper noun" form by determining whether or not the word has an initial capital letter followed by one or more lower case letters. If this condition is met, the syntax code of the word is adjusted in operation 122 to represent the "proper noun" form (NF).

The dictionary contains a number of common words which are likely to lead to a good boundary point in a subtitle. It also contains many common words such as prepositions or articles which do not make good line boundaries. Table 3 gives a list of the types of words found in the dictionary or identified by the WORD routine and the syntax code for each type.

TABLE 3

SYNTAX CODE	SYNTAX CLASS	PART OF SPEECH
nil		word not found in dictionary, so syntax unclassified
0	CC	co-ordinating conjunction
1	SC	subordinating conjunction
2	PP	preposition
3	AR	article
4	NF	confirmed 'proper-noun' form (includes TT)
5	RP	relative pronoun
6	SP	subject case personal pronoun
7	AP	ambiguous case personal pronoun
8	OP	object case personal pronoun
9	DA	demonstrative adjective
10	TT	titles which may be followed by fullstop
11	PP/AV	ambiguous preposition or adverb
12	BE	simple past, present and future of verb 'to be'
13	PA	possessive adjective
14	QA	quantitative adjective
15	AV	adverb (found by disambiguation)

Table 4 gives a sample of some of the words in the dictionary and their syntax classes (those entries marked with a star indicate that although the word is not strictly speaking a part of speech of the type listed it has the same effect as far as suitability for the end of the subtitle line is concerned). The dictionary is stored in strict alphabetical order to speed up the search routine.

TABLE 4

WORD	SYNTAX CLASS	
a	AR	
about	PP, AV	
above	PP, AV	
after	PP, AV	*
all	QA	
am	BE	
an	AR	
and	CC	
any	QA	
are	BE	
as	PP	*
at	PP	
be	BE	
because	SC	
been	BE	
before	PP, AV	*
below	PP, AV	
between	PP, AV	
but	CC	
by	PP, AV	
down	PP, AV	
each	QA	
eight	QA	
every	QA	
except	PP	*
few	QA	
five	QA	
for	PP	
four	QA	
from	PP	
he	SP	
her	PA	*
hers	PN	* (coded PA)
him	OP	
his	PA, PN	* (coded PA)
i	SP	
if	SC	
in	PP, AV	
into	PP	
is	BE	
it	AP	
its	PA	
like	PP, VB	* (coded DA)
many	QA	
me	OP	
mine	PN	* (coded PA)
more	QA	
most	QA	
mr	TT	
mrs	TT	
much	QA	
my	PA	
near	PP, AV	
nine	QA	
none	QA	
of	PP	
off	PP, AV	
on	PP, AV	
one	QA	
or	CC	

TABLE 4 (continued)

WORD	SYNTAX CLASS	
our	PA	
ours	PN	* (coded PA)
out	PP, AV	
over	PP, AV	
seven	QA	
she	SP	
since	PP, AV	*
six	QA	
some	QA	
ten	QA	
than	PP	
that	RP	*
the	AR	
their	PA	
them	OP	
these	DA	
they	SP	
this	DA	
those	DA	
three	QA	
through	PP, AV	
to	PP	
two	QA	
under	PP, AV	
unless	SC	
until	PP	*
up	PP, AV	
upon	PP	
us	OP	
was	BE	
we	SP	
were	BE	
what	RP	
when	SC	
where	SC	
which	RP	
who	RP	
whose	RP	
why	SC	
with	PP	
you	AP	
your	PA	

Figure 10 illustrates the subroutine SEARCH. First in an operation 115 a variable n is set to 1 and then a test 116 is carried out to determine if the ASCII code for the first letter of the word is greater than that for the first letter of the first dictionary word. If this test is satisfied then in an operation 118 the routine moves to the next dictionary entry and re-enters the test 116 by means of a loop 119. If the test 116 is not satisfied then a test 117 is carried out to discover whether the first letter of the word is less than the first letter of the current dictionary word. If this test is satisfied then the search is known to have failed, since the dictionary is stored alphabetically, and the operation 110 of Figure 9 is carried out. However if the test 117 is not satisfied then a condition 120 is established, whereby the ASCII code for the nth letter of the word equals that for the nth letter of the dictionary entry. Since the two codes are equal, a test 123 is carried out to discover whether the currently matched character of the word buffer is a space. If so then a dictionary match has been found and the operation 112 of Figure 9 is carried out. If not then the variable n is incremented by one in an operation 121 and a loop 119 back to the test 116 takes place.

In the operation 112 of Figure 9, the syntax code as given in Table 3 for the part of speech of the matching word is stored and in the operation 113, Table 5 is entered using the syntax code stored and a weight is read out from the first column and this weighting is then stored for use in Task 5.

TABLE 5

WEIGHTS		SYNTAX CLASS	SYNTAX CODE
1	3	CC	0
1	3	SC	1
2	7	PP	2
2	3	AR	3
3	3	NF	4
3	6	RP	5
2	6	SP	6
3	6	AP	7
6	6	OP	8
3	6	DA	9
3	6	TT	10
4	7	PP/AV	11
2	6	BE	12
2	6	PA	13
3	6	QA	14
6	7	AV	15
4	6	default	none

In order to complete Task 0 the subroutine DISAMB is now called in the operation 107 and this subroutine is shown in Figure 11. In a test 126 the syntax code of the current word and the syntax code of the previous word are compared. The following rules for disambiguation are then used wherever the ambiguous syntax class PP/AV occurs in the previous word:-

(a) If the current word is an object case personal pronoun, the syntax code of the previous word is changed to PP.

(b) If the current word is an article, the syntax code of the previous word is changed to PP.

(c) If the current word is a possessive adjective, the syntax code of the previous word is changed to PP.

(d) If the current word is a subject case personal pronoun, the syntax code of the previous word is changed to PP.

(e) If the current word is a preposition, the syntax code of the previous word is changed to AV.

(where the abbreviations used are those of Tables 4 and 5.)

If a match is found then the syntax code of the previous word is changed accordingly in an operation 127 and in an operation 128 the weighting following the previous word is altered accordingly. If no match is found in test 126 then a jump 129 is taken to return from the subroutine.

The next task consisdered is Task 1 which is used to reset previously identified word boundary code to a space character when trailing punctuation occurs signifying that the boundary should come after the punctuation. Since this task is simply a matter of removing the weighting stored in the space before the trailing punctuation no flow diagram is given. An example of circumstances in which Task 1 is carried out is as follows:-

"Harry ? Not likely ! "

The object of Task 2 is to determine punctuation type when punctuation occurs and find a combined punctuation/syntax code using Table 5. In a test 130 (Figure 12) the punctuation is examined to discover whether it is an exclamation mark, a question mark or a fullstop. If not an operation 131 is carried out in which Table 5 column 2 is used to find the weighting appropriate for the end of a clause. If test 130 is positive then a test 132 is carried out to find out whether the punctuation is a fullstop. If it is not then the combined punctuation/syntax weight is given the value 7 in an operation 133 indicating that this would make a very good boundary for the end of a subtitle. If the punctuation is a fullstop then a test 134 is used to discover whether the previous word is a "title", for example Mr., Gen. or Rev., and if not then the weight value 7 is given in the operation 133. Otherwise a weight value of 1 signifying a very poor position for the end of a subtitle line is given in an operation 135.

Since Task 3 simply involves the replacement of the syntax weight stored for the current word by the combined punctuation/syntax weight no flow diagram is given. Task 4 is also a relatively simple task and involves re-setting the following variables to initial values:-

The syntax code for the present word is stored as a syntax code for the last word;

The syntax code for the current word is given a neutral value of zero as is the store for the weight of the current word.

In Task 5 the weight determined is stored in the input buffer in the space after the current word. In some circumstances some types of punctuation such as a comma or a fullstop do not signify a good end of line boundary; for example where a comma occurs in a number such as 3,432 or where a fullstop occurs in a number as a decimal point. In these cases where a number is detected following punctuation Task 6 is carried out to return the punctuation weight to a neutral value.

It will be clear that the invention can be put into practice in many other ways than that specifically described. For example many other forms of apparatus than that shown in Figure 1 are suitable and other algorithms than those described may be used. In the description applicable to Teletext the maximum number of subtitle text and control characters per line is forty but for other applications different line length limits may be used. The text formatting software may also be implemented in other systems not using the Teletext standard, for example in subtitling equipment using direct video character generators.

The syntax analysis and linguistic weighting technique is so structured that it could be readily applied to other languages (for example Dutch) having a similar grammatic structure to English. This can be achieved by substituting the syntax coding (Table 3), the syntax dictionary (Table 4) and the syntax weighting (Table 5) by appropriate tables for the language required.

CLAIMS

1. Apparatus for use in arranging text comprising

means for entering text, and analysis means for storing any text entered, analysing the text stored on the basis of linguistic criteria, and providing signals representing the text including line-end signals indicating positions for the said ends of lines in arranged text,

the line-end signals being derived at least partially on the basis of the said analysis.

2. Apparatus according to Claim 1 for use in preparing subtitles wherein the analysis means is arranged to determine whether a subtitle is too long to be contained in one line and if so to allocate a weighting signal to the space following each word, or the space following any trailing punctuation, in the text stored, the weighting signal being determined from:-

the part of speech of the word, if the analysis means can identify the appropriate part of speech, and/or punctuation, and the position of the word in the subtitle,

and the weighting signals being used in deriving the line-end signals.

3. Apparatus according to Claim 1 wherein the analysis means stores a dictionary of words and the corresponding part of speech of each word in the dictionary, and the analysis means is arranged to analyse the text stored by comparing each word with ech word in the dictionary, and to allocate, each time a match is found, a syntax signal to the space following a word, in the text stored or the space following any trailing punctuation, the syntax signal representing the part of speech of the matching word, the syntax signal being used in deriving the line-end signals.

4. Apparatus according to Claim 1 or 3 wherein the analysis means is arranged to allocate to the space following each punctuation mark which relates to previous text, a punctuation signal dependent on that punctuation mark, the punctuation signal being used in deriving the line-end signals.

5. Apparatus according to Claim 4 wherein the analysis means is arranged to determine whether each punctuation mark relates to text following the mark and if so to allocate a punctuation signal dependent on that punctuation mark to the space preceding the mark.

6. Apparatus according to any of Claims 3, 4 or 5 for use in preparing subtitles wherein the analysis means is arranged to determine whether the text of a subtitle is too long to be contained in one line and if so to allocate to spaces in the text a position signal according to the suitability of the position in a subtitle of respective spaces for the end of a line in the arranged subtitle, the position signals being used in deriving the line-end signals.

7. Apparatus according to Claim 6 wherein the analysis means is arranged to allocate weighting signals to spaces in the text stored, the weighting signals being dependent on the position signals and on any syntax and punctuation signals allocated to respective spaces.

8. Apparatus according to any of Claims 2, 6 or 7 wherein the analysis means includes a store in which signals representing the stored text are held in a sequence corresponding to the order of words in a subtitle, and signals allocated to spaces are stored in the store, each allocated signal being stored in the sequence where the corresponding space occurs in the text.

9. Apparatus according to Claim 2, or any of Claims 5 to 8 wherein the signals allocated are numbers and the magnitude of each number indicates the suitability of the space to which it is allocated to be the end of a subtitle line, and the analysis means is arranged to generate the said line-end signals according to the magnitude of the said numbers.

10. Apparatus according to Claim 9 insofar as dependent on Claim 8 wherein the analysis means is arranged to determine a range or ranges of possible line end positions which allow all the text of a subtitle to be displayed, and to generate line-end signals only according to the magnitude of numbers stored in the sequence in positions corresponding to line ends within the said range or ranges.

11. Apparatus according to Claim 2 or any of Claims 6 to 10 wherein the analysis means includes an output buffer capable of storing a plurality of lines of characters and the analysis means is arranged to so load the output buffer at least partially in accordance with the line-end signals that each character of a subtitle has a position in the buffer which corresponds to its position in a subtitle to be displayed, and wherein the apparatus includes means for recording the contents of the output buffer in a form which allows the said plurality of lines to be recovered.

12. Apparatus according to Claim 11 wherein the means for entering text allows commands relating to at

least one of the following subtitle parameters to be entered: position, colour of characters, colour of background, and the analysis means is arranged to insert control characters representing the parameters to be inserted into the output buffer in positions corresponding to part of a display line without interfering with the text of that line, the control characters being positioned to control the display of that line when the subtitle is displayed.

13. Apparatus according to Claim 12 wherein the analysis means is arranged to insert in the output buffer start and, if space in a line permits, end characters signifying the start and end of a box surrounding a subtitle, after the text of the subtitle has been divided into lines (where necessary) and positioned in the said buffer, the start and end characters being positioned in the buffer before and, if space permits, after, respectively, each subtitle line.

14. Apparatus according to any preceding claim wherein the analysis means comprises a computer programmed to analyse the text stored and generate the line-end signals.

15. A method for use in arranging text comprising storing text, automatically analysing the text on the basis of linguistic criteria, and providing signals representing the text which include line-end signals indicating positions for the ends of lines in arranged text, the line-end signals being derived at least partially on the basis of the said analysis.

16. A method according to Claim 15 for use in preparing subtitles including determining whether a subtitle is too long to be contained in one line and if so allocating a weight to each word in the text stored, the weight depending on the part of speech of the word, and/or punctuation, and the suitability of the position of that word in the subtitle for the end of a line in the subtitle, the weights being used in deriving the line-end signals.

17. Apparatus for use in the preparation of subtitles substantially as hereinbefore described with reference to Figure 1 or 13 and programmed according to any of Figures 2 to 12 or 14.

18. A method of preparing subtitles substantially as hereinbefore described.

Printed for Her Majesty's Stationery Office, by Croydon Printing Company Limited, Croydon, Surrey, 1983.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.